# SECURITY AWARE AND MULTI OBJECTIVE QoS AWARE VIRTUAL MACHINE TREE TASK SCHEDULING & LOAD BALANCING IN CLOUD COMPUTING ENVIRONMENT

A Synopsis Submitted to

**SCHOOL OF ENGINEERING & TECHNOLOGY**

IIMT UNIVERSITY, MEERUT

For the Registration of

**Doctor of Philosophy**

**in**

**Computer Science & Engineering**

**Submitted by**


**Santosh Kumar Maurya**

**(Enrollment Number: IIMTU/2066950)**

**Under the guidance of**

**Dr. Garmia Sinha**

Department of Computer Science & Engineering

IIMT University, Meerut


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**IIMT UNIVERSITY**
**'O' POCKET GANGA NAGAR MEERUT-250001, INDIA**
**2021**

**Signature of Research Scholar**

**Santosh Kumar Maurya**

Research Scholar

Department of Computer Science & Engineering

School of Engineering & Technology

(Reg. No: IIMTU/2066950 Date of enrolment: 21/01/2021)

**Signature of Supervisor**

**Dr Garima Sinha**

Department of Computer Science & Engineering

IIMT University, Meerut

# Brief Profile

---

| | |
|---|---|
| **Tentative Title of Thesis** | **: An Improved Security Aware and Multi Objective QoS Aware Virtual Machine Tree Task Scheduling in Cloud Computing Environment** |
| **Candidate Name** | **: Santosh Kumar Maurya** |
| **Permanent Address** | **: K-154 Khad Pathar Renukoot Sonebhadra** |
| **Official Address** | **: COE office IIMT University, Meerut** |
| **Mobile** | **: 9412444576** |
| **Email** | **: santoshcse200@gmail.com** |
| **Supervisor(s)** | **: Dr. Garmia Sinha** |
| **Date of Registration** | **: 21/01/2021** |

# TABLE OF CONTENTS


| Contents | Page Number |
|---|---|
| **List of Figures** | II |
| **List of Tables** | III |
| **1. Introduction** | 1 |
| 1.1.Introduction | 1 |
| 1.2.Scheduling and Load Balancing | 2 |
| 1.2.1 Load balancing parameters | 3 |
| 1.2.2 Issues in load balancing algorithms | 4 |
| 1.3 Motivation | 5 |
| **2. Literature Survey** | 6 |
| 2.1 Literature Survey | 6 |
| 2.2 Gaps in the Literature | 12 |
| **3. Proposed Research** | 12 |
| 3.1 Problem Definition | 12 |
| 3.2 Objectives | 14 |
| 3.3 Methodology | 14 |
| 3.4 Work Plan | 18 |
| **List of References** | 18 |
| **List of Publications** | 20 |

# LIST OF FIGURES


| Figure No. | Figure Captions | Page No. |
|---|---|---|
| 1.1 | Scheduling and load balancing design | 3 |
| 3.1 | Working set model of proposed algorithm for multi objective QoS | 14 |
| 3.2 | Load balancer | 15 |
| 3.3 | System load | 16 |

# LIST OF TABLES


| Table No. | Table Captions | Page No. |
|---|---|---|
| 2.1 | Latest research publication on load balancing in cloud computing | 11 |
| 3.2 | System-load-output with different fitness values | 17 |

# 1. Introduction

## 1.1 Introduction

Cloud computing is an on-demand services in the computer field to provide the computing environment through the Internet to the clients at any time and on the utility basis. Cloud computing is the extended concept of distributed and parallel computing. Scalability, flexibility, on-demand network access and virtualization are the main features of the cloud environment and all these features are available through Internet. Virtualization is a key concept in cloud computing; it provides a huge execution/computing environment to the cloud users. IaaS(Infrastructure as a service ) , PaaS (platform as a service), and SaaS (software as a service) are three main types of basic services provided by the cloud service providers. Users of the cloud uses the services in utility form (i.e. pay per use basis) and cloud providers offer SLA (service level agreement) and satisfactory level of services to the users. The satisfaction and efficiency of cloud services are always dependent on the effectiveness of the task which has been submitted to the cloud environment by the end user. In order to determine the efficiency, reliability, and performance of cloud services, it is critical that user tasks be scheduled properly and performance the scheduling of tasks is extremely important in this performance analysis. The Cloud computing environment has suffered from many issues, including service availability, security, proper load balancing, resource scheduling, data center energy consumption, and scaling. In the cloud environment balancing the load on the different virtual machines is an important concern. Function of the task scheduler has to assign job/task to virtual machine from available pool of resources and load balancer continuously observing load of each virtual machine, if any virtual machine are under loaded or overloaded then it apply load balancing algorithm to balance the load on each virtual machine and in order to minimize makespan time and increase throughput. The purpose of the load balancing algorithms is to use computer resources properly. So that all virtual machine/nodes can be used as much as possible. It is not that some resources are being used very much and some are not being used. In a cloud computing environment, arrived job/task consists of many interdependent tasks/job and they may run the independent task/job in multiple node/VMs or in the same node/VMs multiple cores. The available heterogeneous computing VMs are handled by assigning the task to appropriate VMs by static or dynamic scheduling to make the cloud computing environment efficient. Proposed algorithm which involves satisfying multiple objectives such as minimizing makespan, cost, reducing task migration during the schedule and enforcing the security are the significant objectives of the

proposed works. With the help of an optimal solution to the cloud scheduling issue, this study effort aims to meet the various needs of cloud users while also minimising costs. Because the cloud is a distributed environment, the scheduling is not limited to a single geographical location. It is possible to schedule the job in a third-party environment as well. As a result, maintaining the security of the job while it is being scheduled in the cloud is another critical factor to consider. Also addressed in this study effort is the issue of delivering security to user-submitted tasks in accordance with the security level chosen by the cloud end user. This is due to the fact that security may actually raise the overhead of the system itself.

## 1.2 Scheduling and Load Balancing

The task matching with VM and load balancing design can be seen in this diagram (figure1). The cloud task manager receives the request from the user, verifies the task's characteristics, such as dependent and independent tasks, and then sends the request to the scheduler. The scheduler uses the scheduling algorithm to determine the most appropriate VM and assign tasks to it. The scheduler places run-time appearance jobs in the most appropriate VMs based on the least-used VM at the moment of appearance.

Task migration occurs in real time based on the demands of the load balancer, with which an overloaded VM can transfer its tasks to an idle VM or one with lesser workload. Whenever it identifies an idle VM or least loaded VM using the resources current state information. The resource monitor talks with all of the VMs' resource probers and gathers information about each VM's capabilities, current load, and number of jobs in the execution/waiting queue. The user provides the task requirement, which includes the length of the tasks to be completed, and the scheduler uses this information to make operational decisions.

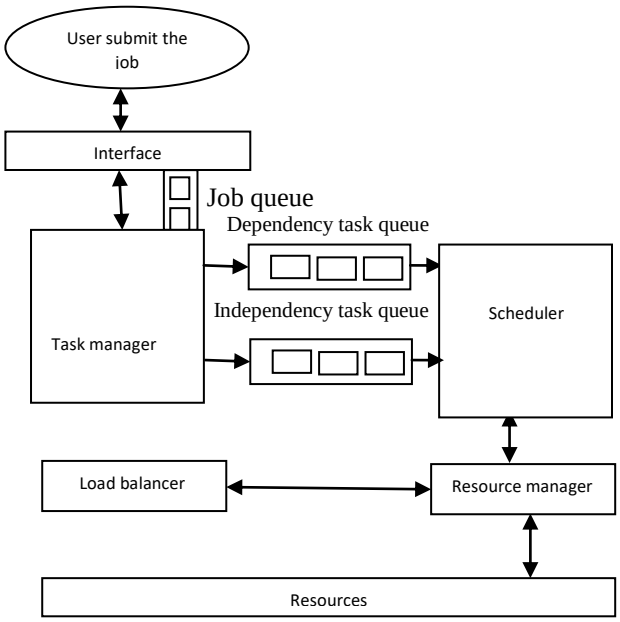


**Figure 1.1:** Scheduling and load balancing design [1]

### 1.2.1 Load balancing parameters

Load balancing parameters in cloud computing. Inside this previously, there are numerous load-balancing techniques presented by academics/researchers [1, 2, 3, 4, 5]. The literature on load balancing provided the following metrics for implementing load-balancing algorithms, which we summarize as follows:

- Throughput: Throughput is the number of jobs accomplished per unit time by a virtual machine virtual machine. A high throughput indicates that the system is doing well. The makespan of a system is inversely proportional to its throughput.

- Response time: It is the sum of the time spent on transmission, waiting, and servicing. As a result, the performance of the system is inversely related to the response time. A faster response time results in a shorter makespan.

- Makespan: It is the time required to complete all tasks submitted to the system in their entirety.
- The system's makespan is the maximum amount of time required for the host to run across the data centre. The optimal makespan results in proper load balancing of the system.

- Scalability: It is a property of a system or model that describes the system's capacity to execute under unusual conditions. It refers to the capacity of a balanced system to survive as the quantity or size of tasks or workload is raised. In a scalable cloud system, resources will be rescaled on a recurring basis.

- Fault tolerance: The fault-tolerant approach is one of the system's capabilities for providing uninterrupted service in the event that one or more system elements fail. Additionally, it removes impediments caused by logical flaws.

  The number of failure points can be used to determine the degree of fault tolerance (i.e., single point failure or multipoint failure).To address some cloud system problems, service providers may require additional resources or virtual machines. However, while this process incurs additional expenditures, the consumer benefits from a fault-free system.

- Migration time: Actual time spent migrating a task or virtual machine from one resource to another.

- The task migration may occur between VMs on the same host or on a separate host. Degree of imbalance: This metric measures the imbalance among VMs.

- Performance: After running a load-balancing algorithm, it calculates the efficiency of the system.

- Energy consumption: It computes the total energy consumption of all nodes. By distributing the load across all nodes, load balancing helps to avoid overheating and thereby reduces energy use.

- Carbon emission: It determines how much carbon is produced by all resources. By transferring loads from underloaded nodes and shutting them down, load balancing plays a vital role in lowering this measure.

### 1.2.2 Issues in load balancing algorithms

[6]There are several difficulties in the cloud environment, including load balancing, data loss, and cloud security. Specifically, we will cover issues pertaining to load balancing, such as how to map forthcoming requests (tasks) to virtual memory in such a way that each task can complete its execution in the shortest amount of time possible. Each load balancing algorithm has a different goal depending on the objective function. For example, some algorithms aim to maximise throughput and average resource utilisation ratio, while others aim to minimize response time, makespan time, and execution time. Neither a conventional algorithm nor a meta heuristic can accomplish all of these goals simultaneously. The performance of the load balancing method is also dependent on the geographical distribution of nodes, their security, and their communication costs. When nodes are dispersed evenly across a defined area, it is

possible to quickly measure fault tolerance and efficiency. If nodes are geographically apart, there is a chance of delay.

Load balancing algorithms in use today might be static or dynamic. Static algorithms are independent of the system's (node's) present state; they are based on prior knowledge (state) about the node's capacity, memory, and user requirements. The primary disadvantage of this sort of algorithm is the rapid failure of system resources and tasks, whereas dynamic algorithms are state-dependent. They outperformed static algorithms, but dynamic algorithms are more complex by nature. [7] Additionally, there are certain flaws with the load balancing mechanism; it is not possible to determine the present state of each node and task. While CPU queue length is a useful predictor of load, measuring it incurs some effort. It's impossible to predict when and who will migrate a task. This is determined by the task migration policy.

- Information policy: Determine what information regarding the workload should be collected, when to begin collecting it, and how to collect it.
- Triggering policy: Define the time period during which load balancing will begin.
- Selection policy: Determine which tasks from an overburdened virtual machine should be transferred to a less-burdened virtual machine.
- Location policy: Identify an appropriate node to which we can transfer the work.
- Transfer policy: Determines whether or not a node is in a condition suitable for task transfer.
- Information policy: This policy determines the status of each node following a specified time interval. If the load on a virtual machine exceeds its capacity, employ elasticity (horizontal or vertical scalability)

## 1.3 Motivation

Cloud computing emerges as a result of the adoption of existing technologies with the goal of lowering costs and simplifying resource administration. Virtualization is the fundamental technology of cloud computing, and it is responsible for ensuring that activities are as active as possible while also ensuring that computer resources are used to their full potential. The capacity=[plm  to operate many virtual computers on a single system is the primary objective

of most significant difficulty they have is maintaining or improving the current level of service during a time of high demand, which is expected to last many months. As a result, despite cloud computing's bright future, there are still a number of important problems that must be addressed before it can be fully realised. Load balancing is also one of the most significant problems. The issue of load sharing arises when several tasks from different users attempt to reach the same server or virtual machine (VM), but the other servers or VMs remain idle due to a lack of requests from the users. This is referred to as a distributed load imbalance system in this scenario. The issue may be resolved by scheduling the request or tasks before they are sent to the server..

As load balancing is very important in a distributed system such as the cloud, grid, and so on. Efficient load balancing methods enable for the efficient distribution of workloads among the available datacenters, servers, hard drives, and other computing resources with a minimum of resources. Job scheduling or load balancing is an issue that involves a number of variables. NP-completeness is widely recognised for this issue, since it cannot be regarded as linear programming in the traditional sense. In order to tackle such an issue, many approximation and branch and bound techniques have been suggested, however the results are poor and require a significant amount of calculation time, which is not feasible in a cloud computing environment. Aside from that, the objective of cloud computing may be too complicated, and it can vary depending on how the company is oriented in the cloud environment, making it difficult to solve in linear time using the current method. In light of the fact that data centre energy consumption has grown to be a significant concern, much research is being conducted in the area of energy-aware scheduling. A common method to saving energy in data centres is to conduct virtual machine consolidation, which is the process of packing virtual machines onto the smallest number of real computers possible. A comparison of conventional scheduling methods and the newly created task scheduling techniques

# 2. Literature Survey

## 2.1 Literature Survey

Numerous cloud load balancing algorithms/techniques have been proposed and developed over the last two decades; some are static [2, 8, 9, 10, 11] while others are dynamic [12, 13, 14].Static algorithms are a subset of load balancing algorithms that require prior knowledge

of node properties, job or task specifications, and cloud resource availability. The static algorithm optimises the QoS metric only when workload variation is low. Dynamic algorithms take into account the current load of each of the computing units (also called nodes) in the system. In this method monitors the virtual machines in cloud data centers continuously.

[1] proposed an improved weighted round robin algorithm that takes the strength of each node/VM and the task size of each arrived job into account when allocating jobs to the most suitable nodes/VMs.Three distinct steps are used in IWRR algorithms to handle the three distinct conditions that occur during the task life cycle.The static scheduler algorithm prioritises initial job delivery, allocating job requests evenly among participating VMs based on the node/strengths VM's and the duration of the requested job.The dynamic scheduler considers the load on all configured nodes/VMs and calculates the approximate completion time of the current load in conjunction with the estimated completion time of the arrived job in each of the participating nodes/VMs.Following that, the minimum possible completion time for this particular job has been determined using the above calculations in one of the nodes/VMs, and the job has been assigned to this VM. The response time is the primary quality of service (QoS) parameter for this algorithm.

[2] proposed an improved min-min (PA-LBIMM) algorithm that decrease the makespan time and increase the resource utilization compare to the min-min algorithm. He used min-min algorithm for initial placement of task on virtual machine. In second phase he selects the smallest size task form heavy loaded machine and calculates the completion time on other virtual machine. Then the Min-Min algorithm makespan time compared with this minimum completion time. If it is less than; job assigned to the virtual machine that produced it.

[11] In this paper writers used credit based scheduling algorithm in cloud computing environment to optimize resources utilization and reduce the makespan. He assigned credit to each task on the basis of task length and priority. Task with highest credit will be assigned to the resources first. He simulates the algorithm on CloudSim simulator and compares the result with algorithm based on task length and task priority and claim that his algorithms perform better. Algorithm does not show better result when number of task increases.

[15] Introduced a heuristic based algorithm for heterogeneous environment in cloud computing. In this algorithms task and VMs both are sorted in descending order of length and capacity. He used FCFS to assign the task to VMs. He defined threshold limit of VMs, If it is

operating at less than 25% of capacity, it is considered underloaded and if utilization more than 85% then it is overloaded.He also do comparative analysis between the proposed load balancing algorithm , FCFS, Min-Min and SJF algorithm. The simulation is repeated over 100 times to determine the makespan. and he wins the competition by getting better results. Hence it provides good load balancing and took less makespan. Simulating tool used was CloudSim.

[17] The purpose of this paper is to develop a genetic algorithm-based load balancing strategy for Cloud Computing in order to maximise resource utilisation in a cloud environment. The analysis of the results indicates that the proposed load balancing strategy not only outperforms a few existing techniques, but also ensures that the customer job meets the required QoS.While it has been assumed that all jobs have the same priority, this may not be the case, this can be facilitated in the job unit vector (JUV) and then taken into account in the fitness function.Additionally, a very straightforward GA approach was used, but variations in the crossover and selection strategies could be used in future work to obtain more efficient and tuned results.

[18] That aims at task/job scheduling with minimum makespan time and minimum cost. CloudSim1 is employed to carry out and simulate the tasks assignment algorithm, and distributed task scheduling. The outputs are compared with Activity based costing algorithm. The conclusion is that the scheduling algorithm employed is better than Activity based costing algorithm.

[19] They developed a load balancing technique for cloud computing environments based on the forage for food strategy of honey bees. This algorithm not only balances the load, but also takes the priority of tasks removed from heavily loaded Virtual Machines into account. The tasks that are removed from these virtual machines are treated as honey bees, which are the global information updaters. Additionally, this algorithm takes into account the tasks' relative importance. Load balancing inspired by honey bee behavior increases overall processing throughput, while priority-based balancing focuses on minimizing the amount of time a task spends waiting on a VM's queue. As a result, the response time of virtual machines is decreased.

[21] Priority Based Job Scheduling using IBA and EASY backfilling algorithm is proposed to resolve the problems which exist in the other backfilling and job scheduling strategies. As per simulation results, They compared to other algorithms the proposed algorithm enhanced the

resource utilization. This work mainly discusses about, a job scheduling algorithm based on priority, IBA and EASY backfilling strategy for cloud computing. It integrates the advantages of priority and backfilling strategy, therefore increases the job scheduling efficiency of the system to a certain extent, as well as systems resource utilization rate.

[23] Similar to the Throttled algorithm, the modified throttled algorithm maintains an index table of virtual machines and also their state.An attempt has been made to improve response time and maximise the utilisation of available virtual machines. In this paper, an efficient approach to handle the load at servers by considering both availability of VMs for a given request and uniform load sharing among the VMs for the number of requests served. The work aimed at efficient method for load balancing, depicted from its two different objectives. One being the response time required to serve the requests and other being the distribution of load among the existing VMs. When compared to existing Round-Robin and Throttled algorithms, the response time for proposed algorithm has improved considerably. Proposed algorithm distributes load nearly uniform among VMs, with improved response time compared to existing algorithms.

[24] Aim of the ant colony optimization to search an optimal path between the source of food and colony of ant on the basis of their behavior. This approach aims efficient distribution of work load among the node. When request is initialized the ant starts movement towards the source of food from the head node. He proposed a modified approach to ant colony optimization that has been applied to cloud or grid network systems with the primary goal of node load balancing. The way in which ants are created and their functionalities according to the pheromones trails and node encountered is clearly visible from the segment of pseudocode provided in the appendix. This modified algorithm outperforms the original approach, in which each ant constructs their own unique result set, which is then combined into a complete solution.Rather than updating their own result set, the ants continuously update a single result set. In this way, the solution set is gradually built on and continuously improved upon rather than being compiled only once in a while. The other advantage of the approach lies in the fact that the task of each ant is specialized rather than being general and the task depends on the type of first node that was encountered whether it was overloaded or underloaded.

[25] It is a static load balancing algorithm (Opportunistic Load Balancing Algorithm) it does not take into account each system's current workload.As a result, it keeps each node busy by

randomly assigning unfinished tasks to available nodes.As a result, the algorithm performs poorly when it comes to load balancing.It fails to calculate the node's implementation time, lowering the processing task's performance.Additionally, when nodes are idle, the cloud system will experience bottlenecks.

[26] This is a dynamic load balancing algorithm that determines which virtual machines are least loaded and assigns them a load. Load balancing controllers keep track of all servers and requests in the server's index table. As a result, when a new request is received, the data centre anticipates that the index table will identify the servers that are least loaded or idle. That is, the algorithm assigns load to servers in a first come, first served fashion. The task is identified by its server-id, and when the server receives a load, its state is updated in the index table. Similarly, when a task is completed, the data centre and controllers receive the information, which reduces the index table's server state. When an internet user submits a request, the load balancer recursively scans the index table and assigns processes appropriately.

[27] Suggested alternative Whale's technique is employed in load balancing in cloud computing. The first thing done in this paper is explain how balanced loads are handled by cloud computing. Secondly, a whale optimization algorithm is used to increase population diversity, which relies on the Sine and Cosine Algorithm for local optimization and the Leapfrog Algorithm for individual screening. Next, a simulation was run to demonstrate that the algorithm proposed in this paper is better at performing tasks, has higher fairness and energy conservation than previous algorithms (such as the ant colony algorithm, particle swarm algorithm, and whale optimization algorithm).

[28] In this paper, a new Spider Monkeys algorithm proposed for load balancing called Spider Monkey Optimization Inspired Load Balancing (SMO-LB) based on mimicking the foraging behavior of Spider Monkeys. It aims to balance the load among virtual machines to increase the performance by reducing makespan and response time.

The variety of latest load balancing work is detailed through Table 2.1 mentioned here. The table will explore the findings of existing load balancing techniques that will be viable for future papers as a valuable investigation to achieve their research objective.

Table 2.1 Latest research publication on load balancing in cloud computing

| S. No. | Ref | Type of the paper | Proposal/Technique | Year | Publisher | Outcome |
|---|---|---|---|---|---|---|
| **1** | [32] | Review paper | - | 2019 | Springer | Detailed encyclopedic review for load balancing techniques |
| **2** | [33] | Technical original paper | VM Allocation based on their Processing Power | 2020 | Other | Reduction in user request response time and data centre processing time |
| **3** | [34] | Review paper | - | 2019 | Other | Study of various hybrid load balancing algorithms in cloud computing environment |
| **4** | [35] | Technical original paper | Energetic neighbor selection for scheduling and transmission | 2019 | Other | Improved network performance and energy efficiency in terms of throughput, packet delivery, delay, packet loss and remaining energy |
| **5** | [36] | Technical original Article | Adding more nodes without the need to increase the power of each node | 2019 | Other | Reduction in throughput and response time |
| **6** | [37] | Technical original Article | Randomization and greedy load balancing algorithm | 2019 | Other | Reduction in cost, response time and data processing time |
| **7** | [38] | Technical original Article | Throttled Modified Algorithm | 2018 | Other | Reduction in response times and processing time |
| **8** | [39] | Review paper | Developed an effective load reconciliation algorithmic rule mistreatment divisible load programming theorem | 2017 | Other | Shown improved performance in terms of throughput and latency |
| **9** | [40] | Review paper | - | 2018 | Other | Shown optimized load on cloud servers along with a<br>proposed work based on clustering algorithm |
| **10** | [41] | Review paper | Particle Swarm<br>Optimization | 2019 | Other | Compare a particle swarm optimization<br>technique with round robin, Ant colony and<br>honeybee foraging load balancing algorithm |
| **11** | [42] | Technical original Article | Genetic algorithm is improved for VM migration | 2017 | Other | Reduced the execution time-space and bandwidth utilization |
| **12** | [43] | Review and proposal | Problems and need for a novel balancing algorithm that employs fault tolerance metrics | 2020 | IEEE Access | Critical study of existing techniques of LB with various parameters such as throughput, performance, migration time, response time, overhead, resource usage, scalability, fault tolerance and power savings |
| **13** | [43] | Review paper | Various standard load balancing<br>techniques & challenges | 2019 | Other | better utilization and better understand for users' needs |
| **14** | [44] | Review paper | Various Load Balancing techniques in a static, dynamic, | 2021 | Elsevier | Addressed the Data Center Response Time and overall performance |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | and nature-inspired | | | |
| **15** | [45] | Review paper | Taxonomy for the load balancing algorithms | 2018 | Elsevier | Analyzed the performance of heuristic-based algorithms through CloudSim |
| **16** | [46] | Review paper | - | 2020 | Other | Shown the matrices that affect the load balancing algorithm |
| **17** | [47] | Review paper | - | 2020 | Other | detailed study on the different load balancing algorithms |
| **18** | [48] | Review paper | - | 2019 | IEEE | Surveyed the comparison and contrast of various loads balancing policy |
| **19** | [49] | Review paper | - | 2019 | Springer | Shown the concept for reducing energy consumption and response time |
| **20** | [50] | Review paper | - | 2019 | Other | Shown conceptual model for efficient resource management & reliability for cloud services |
| **21** | [51] | Review and proposal | Concentrate on scheduling at the local and global level for optimal use of computing elements | 2018 | Other | Proposed method (Decider, Threshold based), with various parameters such as RAM, CPU consumption, network traffic statistics |
| **22** | [52] | Review paper | Load balancing techniques categorized | 2021 | Elsevier | Hidden Markov Model helps in better CPU utilization as well as faster execution environment |
| **23** | [53] | Review paper | - | 2018 | Springer | Enhanced the efficiency and reliability |
| **24** | [54] | Review paper | Stack modifying estimation | 2021 | Other | Resource utilization |

With the goal in mind, the literature has been reviewed, and a proposal for an algorithm has been developed.

## 2.2 Gaps in the Literature

A review of the literature or studies that gave rise to the notion that further research in the direction of a Load Balancing algorithm could be conducted. In most cases, researchers/authors have not considered optimising all the QoS parameters; they have only optimised one or two QoS parameters. In most cases, they have not considered the security of the task and virtual machine during task mapping.

# 3. Proposed Research

## 3.1 Problem Definition

A requirement in cloud computing for the maximum or efficient use of computer resources in cloud platforms, which results in a return on investment that is favourable to the client from

the customer's point of view, may be called efficient resource management. Because of the availability of a wide solution space and the length of time required to arrive at an optimum solution in a cloud computing environment, load balancing or task scheduling in cloud computing environments falls into the category of NP-hard problems. n tasks and m Virtual Machines (VMs) are used in cloud computing for load balancing to get optimal performance. Each job should be completed by each of the m VMs in order to reduce the overall processing time. Scheduling algorithms are primarily used to reduce the number of QoS parameters that must be fulfilled. Task Scheduling is responsible for dealing with the situation in which resources must be allocated to a newly acquired task. One of the most effective scheduling algorithms takes into account balancing out the load on the system while accounting for the entire execution time of available resources, resulting in the reduction of execution cost to a minimum.

The following goals have been established for this research project in order to be successful.

A. It is important to identify and analyse the existing approaches for efficient load balancing or job scheduling in a cloud computing environment before developing new ones.

B. Utilizing the most appropriate soft computing techniques, we aim to design an efficient task scheduling system with security that will allow us to accomplish resource optimization in cloud computing environments. My proposed model is given in figure 3.1.

**C.** Evaluation and comparison of the task scheduling methods that have been created.

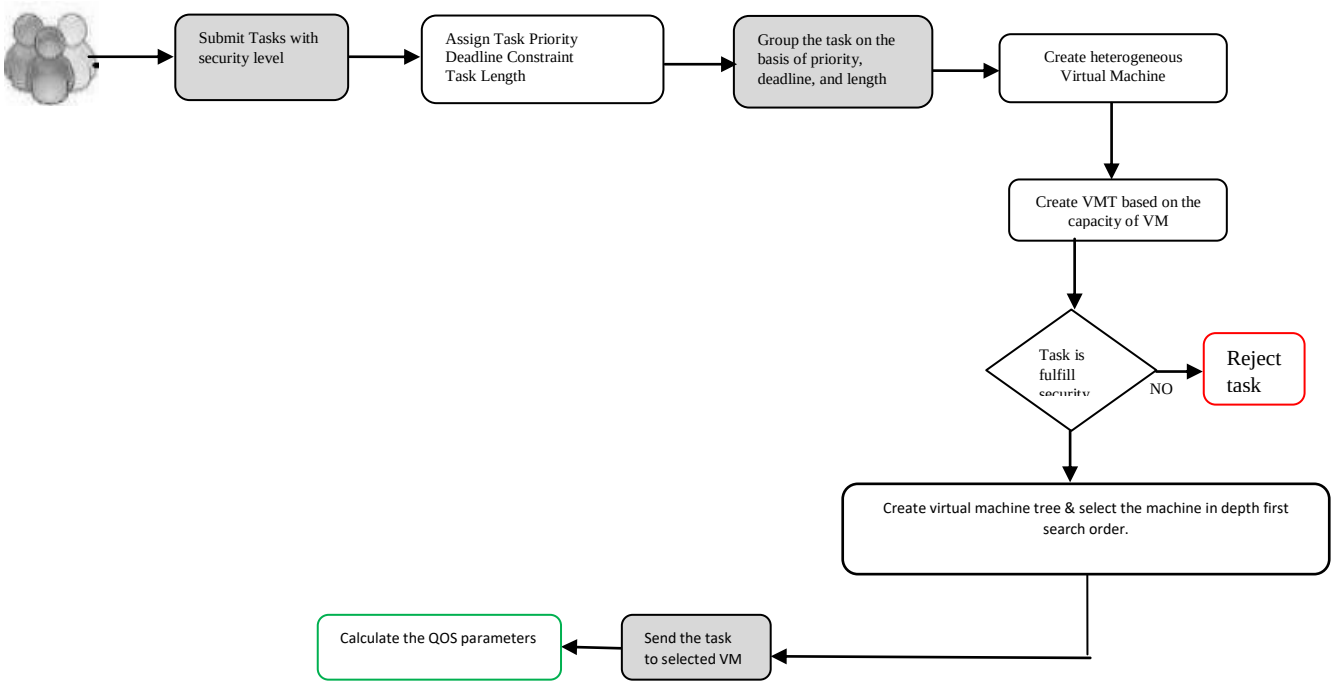
Submit Tasks with security level
Assign Task Priority
Deadline Constraint
Task Length
Group the task on the basis of priority, deadline, and length
Create heterogeneous Virtual Machine
Create VMT based on the capacity of VM
Task is fulfill
NO
Reject task
Create virtual machine tree & select the machine in depth first search order.
Send the task to selected VM
Calculate the QOS parameters

Figure 3.1: Working set model of proposed algorithm for multi objective QoS

## 3.2 Objectives

The goal of this research is to increase the number of completed tasks with a deadline and other QoS parameters, as well as the cost of resource utilisation in a heterogeneous environment. The research will use data protection technologies that provide various levels of security with varying overheads for data transfers and task executions in public clouds. We present a technique security oriented virtual machine tree (SOVMT). SOVMT use optimization techniques to minimise QoS parameters.

## 3.3 Methodology

A tree mechanism for balancing the load is shown in most of the papers. Because a tree tries to balance the load on either side of the center of the root. Agreeing with the tree concept, some other theories are also capable of doing the distribution of work with equitable division such as decision theory. Continuing with the proposed concept, it operates with two variables, either 1 or 0. With large amounts of data, applications are picked up from a pool of applications. The nature or basics of each application are already determined at the time of incoming requests or registration of applications. The predefined system automatically rejects applications if it is found to be inappropriate for the configured system. And then a new machine is quickly setup to take that into account, as the application requires. The objective

of the innovation system is to meet all incoming and registered requests as soon as possible so as to achieve useful results for the development of new technology or methods.

Figure 3.2 serves as a landscape view. Many applications are to be processed. A load balancer will allocate applications to different servers according to its configuration and capacity. A greater number of applications would be needed to accommodate the highly capable servers. The execution process will process the shorter length task with the lower configured system. Sometimes requests have to wait for execution or the machine is not found, this usually happens when there is no space in the machine to execute more applications. Then a threshold value is to be set which will indicate that it is either 0 or 1. If the server is close to reaching the ideal state, it will be 1 and show that the machine is still working and there is no room for more application. But if the value reaches 0 in a very short time, this means, now the machine is able to run the next application. The length of application (MI) will be the only factor to calculate the size of the application machine usage count (MUC). But the system configuration will be measured with the help of various parameters like bandwidth Mbps, memory (usually GB), size instruction count (IC) and speed million instructions per second (MIPS). In order to obtain the best mapping of applications to the appropriate machine, the fitness measured must be in the calculation.

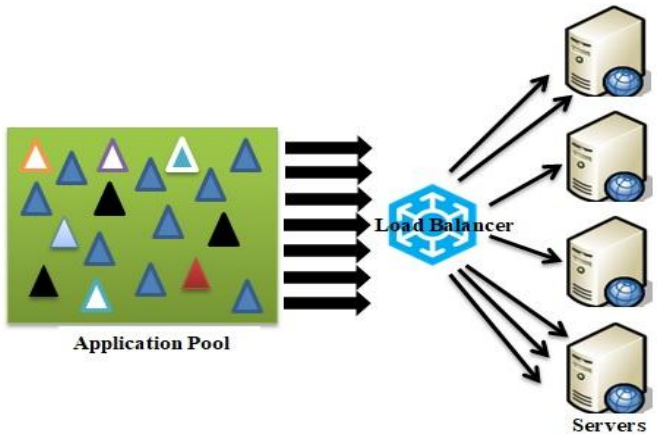


Figure 3.2. Load balancer

An attempt is made to propose a decision-based load-balancer (DLB) model in the paper. And it also tries to address the gaps, which are generally covered in current research papers. Such references are already described above in existing concepts.

A stable system is recognized by its functionality. It should achieve 100% accuracy in the results. Still, no system is capable of performing 100% accurate. With regard to this issue a process is being initiated through DLB to get the best results with as little as possible. The proposed system will show answer in only two states i.e., 1 or 0. A range value will only be set between 0 and 1. When the system is in the ON state, the value will be initially set to 1 and the variation will be noted until the 0 value is reached.

For better performance a fitness value must be calculated at every moment (mainly: new application come and in waiting state) to check the load state of a system. Mapping of n number of applications should be allotted to the appropriate and ideal machine. A counter has to be set to monitoring the queue for coming applications. If counter found application in a queue, then fitness count will be measured of the system and if the value found 0 then the coming application will get executed. Top layer in Figure 3.3 shows that this time system is fully occupied with the load. And in same figure, lower stages are showing that system is now available to take more applications into account as it reached to 0.

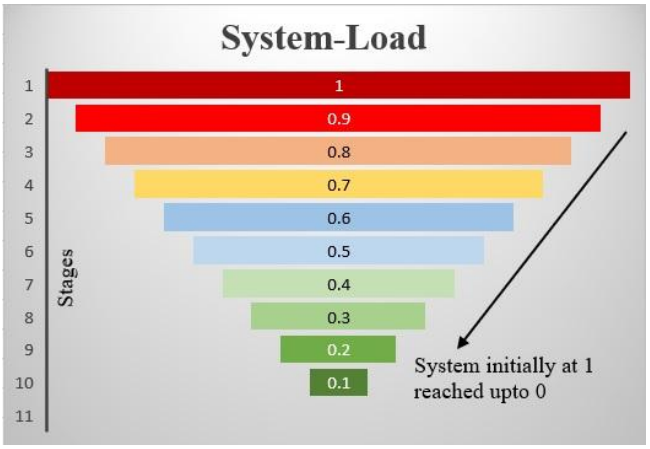


Figure 3.3. System load

Initially fitness value variable introduced as (FV) and initially counter C is set to be at 0 value. An application came into queue, counter C will get update its value as 1. Now FV will measure the machine state value from 1 to 0. Table 2 summarizes every possible system stage with its outcome.

To calculate the fitness value for ith application and jth machine, an expected executed time (EET) has to be measured for a single application/process. Then it will collaborate with the decision value (DV), and finally the proposed system will be into execution.

DVij = = System stage is 11 (Specified) (1)

$$FV=\sum_{i=0}^{n} DV_{ij} \times EET_{ij} \quad (2)$$

Let's say FV will get the value near any stage then the result will be found in the table. Using (1) and (2), the value will definitely go from 1 to 0 because the pre-requisites of EET are to be measured by measuring the length of the task before the start of execution.

In recent studies, several load-balancers have been taken into account. However, they all have some benefits and some limitations as well. Some are only capable of balancing the network load and some are only for hardware load balancing. So, the balancer introduced in the past was specified only with certain specified conditions. The concept of decision-based load-balancer enables balancing of loads with multiple requirements with the help of decision and fitness theory.

Table 3.2. System-load-output with different fitness values

| System stages | Counter | Fitness value | Outcome |
|---|---|---|---|
| 1 | 1 | 1 | no allocation |
| 2 | 1 | 0.9 | no allocation |
| 3 | 1 | 0.8 | no allocation |
| 4 | 1 | 0.7 | no allocation |
| 5 | 1 | 0.6 | no allocation |
| 6 | 1 | 0.5 | no allocation |
| 7 | 1 | 0.4 | no allocation |
| 8 | 1 | 0.3 | no allocation |
| 9 | 1 | 0.2 | no allocation |
| 10 | 1 | 0.1 | no allocation |
| 11 | 1 | 0 | Allocation |
| - | 0 | - | - |

## 3.4 Work Plan

| Activity / Months | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 | 22 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Literature Survey & Develop Questionnaire | | | | | | | | | | | | |
| Review of load balancing models of different category and analysis of data | | | | | | | | | | | | |
| Identification of areas of Improvements | | | | | | | | | | | | |
| Enhancement of model of Load Balancing | | | | | | | | | | | | |
| Evaluation and Comparison | | | | | | | | | | | | |
| Testing | | | | | | | | | | | | |
| Thesis Writing | | | | | | | | | | | | |

# References


[1] D. Chitra Devi and V. Rhymend Uthariaraj , "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks", Volume 2016 |Article ID 3896065 | https://doi.org/10.1155/2016/3896065

[2] Huankai Chen, F. Wang, N. Helian and G. Akanmu, "User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing," 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH), 2013

[3] Kumar, P., & Kumar, R. (2019). Issues and challenges of load balancing techniques in cloud computing: a survey. ACM Computing Surveys (CSUR), 51(6),1-35.

[4] Hota, Arunima & Mohapatra, Subasish & Mohanty, Subhadarshini. (2019). Survey of Different Load Balancing Approach-Based Algorithms in Cloud Computing: A Comprehensive Review. 10.1007/978-981-10-8055-5_10.

[5] Sambit Kumar Mishra, Bibhudatta Sahoo, Priti Paramita Parida,Load balancing in cloud computing: A big picture,Journal of King Saud University - Computer and Information Sciences,Volume 32, Issue 2,2020,Pages 149-158

[6] KUMAR, MOHIT. (2013). A Technique to reduce the Economic Denial of Sustainability (EDoS) Attack in Cloud.

[7] KUMAR, MOHIT. (2018). Load balancing algorithm to minimize the makespan time in cloud environment.

[8] Atul Vikas Lakra, Dharmendra Kumar Yadav, Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput optimization,Procedia Computer Science,Volume 48,2015.

[9] Li, Ji & Feng, Longhua & Fang, Shenglong. (2014). An Greedy-Based Job Scheduling Algorithm in Cloud Computing. Journal of Software. 9. 10.4304/jsw.9.4.921-925.

[10] P. Samal,P. Mishra, "Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing", International Journal of computer science and Information Technologies, Vol.4, No.3, pp.416-419, 2013.

[11] Thomas, A., Krishnalal, G., Raj, V. P. J. Credit based scheduling algorithm in cloud computing environment. Procedia Computer Science, 2015, 46: 913–920.

[12] Espadas J., Molina A., et al. A tenant-based resource allocation model for scaling;software-as-a-service applications over cloud computing infrastructures. Future Generation Computer Systems, 2013, 29(1): 273–286.

[13] Kumar, Mohit and Sharma, S. C. Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment. Computers & Electrical Engineering, 2017.

[14] Somasundaram, Thamarai Selvi & Kannan, Govindarajan. (2014). CLOUDRB: A framework for scheduling and managing High-Performance Computing (HPC) applications in science cloud. Future Generation Computer Systems. 34. 47 - 65. 10.1016/j.future.2013.12.024.

[15] KUMAR, MOHIT & Sharma, Subhash. (2017). Dynamic load balancing algorithm to minimize the makespan time and utilize the resources effectively in cloud environment. International Journal of Computers and Applications.

[16] Omara, Fatma & Abdelkader, Doaa. (2012). Dynamic Task Scheduling Algorithm with Load Balancing for Heterogeneous Computing System. The Egyptian Informatics Journal. 13. 135-145. 10.1016/j.eij.2012.04.001.

[17] Dasgupta, Kousik & Mandal, Brototi & Dutta, Paramartha & Mandal, Jyotsna & Dam, Santanu. (2013). A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing. Procedia Technology. 10. 340–347. 10.1016/j.protcy.2013.12.369.

[18] Selvarani, Shaila & Sadhasivam, Sudha. (2010). Improved cost-based algorithm for task scheduling in cloud computing. Computational Intelligence and Computing Research (ICCIC). 10.1109/ICCIC.2010.5705847.

[19] Dhinesh Babu L.D., P. Venkata Krishna,Honey bee behavior inspired load balancing of tasks in cloud computing environments,Applied Soft Computing,Volume 13, Issue 5,2013,

[20] Wong, Adam & Goscinski, Andrzej. (2007). Evaluating the EASY-backfill job scheduling of static workloads on clusters. 64 - 73. 10.1109/CLUSTR.2007.4629218.

[21] Dubey, Kalka & KUMAR, MOHIT & Chandra, Mayank. (2015). A Priority Based Job Scheduling Algorithm Using IBA and EASY Algorithm for Cloud Metaschedular.

[22] Mishra, Ratan & Jaiswal, Anant. (2012). Ant colony Optimization: A Solution of Load balancing in Cloud. International journal of Web & Semantic Technology. 3. 10.5121/ijwest.2012.3203.

[23] S. G. Domanal and G. R. M. Reddy, "Load Balancing in Cloud Computing using Modified Throttled Algorithm," 2013 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2013, pp. 1-5, doi: 10.1109/CCEM.2013.6684434.

[24] K. Nishant et al., "Load Balancing of Nodes in Cloud Using Ant Colony Optimization," 2012 UKSim 14th International Conference on Computer Modelling and Simulation, 2012, pp. 3-8, doi: 10.1109/UKSim.2012.11.

[25] "Cloud Load Balancing Image,".[Online].

[26] Rai, S., Sagar, N., & Sahu, R. (2017). An Efficient Distributed Dynamic Load Balancing Method based on Hybrid Approach in Cloud Computing. International Journal Of Computer Applications, 169(9), 16-21. doi: 10.5120/ijca2017914876

[27] Xiaoxia Li. (2021). A Study into Load Balancing in Cloud Computing Based on Whale Optimization Algorithm. CONVERTER, 2021(7), 180-189. Retrieved from http://www.converter-magazine.info/index.php/converter/article/view/487

[28] Alshattnawi, Sawsan, and AL-Marie Mohammad. "Spider Monkey Optimization Algorithm for Load Balancing in Cloud Computing Environments."

[29] R. B. Bohn, J. Messina, F. Liu, J. Tong and J. Mao, "NIST Cloud Computing Reference Architecture," IEEE World Congress on Services, 2011, pp. 594-596, doi: 10.1109/SERVICES.2011.105.

[30] Nidhi Bansal and Dr. Maitreyee Dutta, Performance Evaluation of Task scheduling with priority and non-priority in cloud computing," International Conference on Computational Intelligence and Computing Research, IEEE, 2014.

[31] Santanu Dam, Gopa Mandal, Kousik Dasgupta and Paramartha Dutta, "Genetic algorithm and gravitational emulation-based hybrid load balancing strategy in cloud computing," Third IEEE International Conference on Computer, Communication, Control and Information Technology (C3IT), pp. 1–7, 2015.

[32] L. Shahbaz Afzal and G. Kavitha, "Load balancing in cloud computing – A hierarchical taxonomical classification," Journal of Cloud Computing: Advances, Systems and Applications, vol. 8, no. 22, 2019, doi: 10.1186/s13677-019-0146-7.

[33] Badshaha P Mulla1, C. Rama Krishna and Raj Kumar Tickoo, "Load balancing algorithm for Efficient VM Allocation in Heterogeneous Cloud," International Journal of Computer Networks & Communications (IJCNC), vol.12, no.1, 2020, doi: 10.5121/ijcnc.2020.12106.

[34] Chukwuneke Chiamaka Ijeoma, Inyiama, Hyacinth C.; Amaefule, Samuel, Onyesolu, Moses Okechukwu and Asogwa, Doris Chinedu, "Review of Hybrid Load Balancing Algorithms in Cloud Computing Environment,," International Journal of Trend in Research and Development (IJTRD), vol. 6, no. 6, 2019.

[35] K. Venkatachalapathy and D. Sundaranarayana, "A Min-Max Scheduling Load Balanced Approach to Enhance Energy Efficiency and Performance of Mobile Adhoc Networks," International Journal of Computer Networks & Communications (IJCNC), vol. 11, no. 3, 2019.

[36] A Eman Daraghmi and Yousef-Awwad Daraghmi, "Advanced Diffusion Approach to Dynamic Load-Balancing For cloud Storage," International Journal of Distributed and Parallel Systems (IJDPS), vol. 10, no. 2/3, 2019.

[37] Dina Farouk Altayeb and Fatima Abdelghani Mustafa, "Analysis On Load Balancing Algorithms Implementation on Cloud Computing Environment," International Journal of Innovative Research in Advanced Engineering, vol. 6, pp. 32-36, 2019, doi: 10.26562/IJIRAE.2019.FBAE10081.

[38] P Nguyen Phi, Cao Trung Tin, Luu Nguyễn and Cong Hung, "Proposed Load Balancing Algorithm to Reduce Response Time and Processing Time on Cloud Computing," International Journal of Computer Networks and Communications, vol: 10, no. 3, pp. 87-98, 2018, doi: 10.5121/ijcnc.2018.10307.

[39] Harshdeep Sharma and Gianetan Singh Sekhon, "A Review on Load Balancing in Cloud Using Enhanced Genetic Algorithm," International Journal of Computer Engineering & Technology (IJCET), vol. 8, no. 2, pp. 23–29, 2017.

[40] Santosh Kumar Upadhyay, Amrita Bhattacharya, Shweta Arya and Tarandeep Singh, "Load Optimization in Cloud Computing using Clustering: A Survey," International Research Journal of Engineering and Technology (IRJET), vol. 5, no. 4, 2018.

[41] Revathi S, Aniz Rizwan and Anusha N, "Optimization of load balancing in cloud using Swarm intelligence: A survey," International Journal of Current Engineering and Scientific Research, vol. 6, no. 6, 2019, doi: 10.21276/ijcesr.2019.6.6.29.

[42] Harshdeep Sharma and Gianetan Singh Sekhon, "Load Balancing in Cloud using Enhanced Genetic Algorithm," International Journal of Innovations & Advancement in Computer Science, vol. 6, no. 1, 2017.

[43] Muhammad Asim Shahid, Noman Islam, Muhammad Mansoor Alam, Mazliham Mohd Su'ud and Shahrulniza Musa, "A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach," IEEE Access, vol. 8, 2020.

[44] K. Samunnisa, G. Sunil Vijaya Kumar and K. Madhavi, “A Circumscribed Research of Load Balancing Techniques in Cloud Computing,” Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP), vol. 8, no. 6S4, pp. 336-339, 2019.

[45] Dalia Abdulkareem Shafiq, N.Z. Jhanjhi and Azween Abdullah, “Load balancing techniques in cloud computing environment: A review,” Journal of King Saud University - Computer and Information Sciences, 2021, doi: 10.1016/j.jksuci.2021.02.007.

[46] Sambit Kumar Mishra, Bibhudatta Sahoo and Priti Paramita Parida, “Load balancing in cloud computing: A big picture,” Journal of King Saud University - Computer and Information Sciences, vol. 32, no. 2, 2020, doi: 10.1016/j.jksuci.2018.01.003.

[47] Pooja Arora and Anurag Dixit, “An optimized Load Balancing Algorithm in Cloud Computing,” Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP), vol. 9, no. 5, pp. 683-688, 2020.

[48] Ankita Jaiswal and Aisha Fernandes, “Load Balancing Algorithms for Cloud Computing Environment,” Journal of Emerging Technologies and Innovative Research, vol. 7, no. 5, pp. 59-67, 2020.

[49] A. Jyoti, M. Shrimali and R. Mishra, "Cloud Computing and Load Balancing in Cloud Computing -Survey,” International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 51-55, 2019, doi: 10.1109/CONFLUENCE.2019.8776948.

[50] Shetty, S.M. and Shetty, S., “Analysis of load balancing in cloud data centers,” J Ambient Intell Human Comput, 2019, doi: 10.1007/s12652-018-1106-7.

[51] Basetty Mallikarjuna and Arun Kumar Reddy Doddi, “The Role of Load Balancing Algorithms in Next Generation of Cloud Computing,” Journal of Advanced Research in Dynamical and Control Systems, vol. 11, no. 7, 2019.

[52] Stephen. A and Daks Ravindran, “Survey on Load Balancing in Cloud Computing,” International Journal of Scientific Research in Computer Science Applications and Management Studies, pp. 37-47, 2018, doi: 10.1109/CCEM.2015.31.

[53] Nidhi Bansal and A.K. Singh, “Grey Wolf Optimized Task Scheduling Algorithm in Cloud Computing,” In: Satapathy S., Bhateja V., Nguyen B., Nguyen N., Le DN. (eds) Frontiers in Intelligent Computing: Theory and Applications. Advances in Intelligent Systems and Computing, vol 1013, 2020. Springer, Singapore. https://doi.org/10.1007/978-981-32-9186-7_16.

[54] Nidhi Bansal, A. K. Singh, “Trust for task scheduling in cloud computing unfold it through fruit congenial,” In: Networking Communication and Data Knowledge Engineering, vol. 04, 2017, pp. 41–48.

[55] Ananya Jain, “Study on Load Balancing Algorithms and Performance Metrics in Cloud Computing,” International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 7, 2021.

# LIST OF PUBLICATIONS

[1] Santosh Kumar Maurya, Dr. Garmia sinha, " Load Balancing in Cloud Computing: An Analytical Review and Proposal", Indonesian Journal of Electrical Engineering and Computer Science (p-ISSN: 2502-4752, e-ISSN: 2502-4760) indexed by **Scopus** (http://ijeecs.iaescore.com) May-2022

[2] Santosh Kumar Maurya, Dr. Garmia sinha, "ANALYSIS OF LOAD BALANCING ALGORITHMS IN THE CLOUD COMPUTING ENVIRONMENT: A REVIEW", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162, Vol.8, Issue 11, page no. ppb235-b242, November-2021